

APPENDIX**IN THE SPECIFICATION:**

Page 1, lines 9-12:

U.S. Provisional Patent Application Serial No. 60/160,759 entitled USE OF A JAVA VM INSTANCE AS THE BASIC UNIT OF USER EXECUTION IN A SERVER ENVIRONMENT, filed on October 21, 1999 by Harlan Sexton *et al.*[, (docket 50277-325; OID-1997-048-19PRO)];

Page 1, line 13-15:

U.S. Provisional Patent Application Serial No. 60/185,136 entitled MEMORY MANAGEMENT USING MIGRATION FOR A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.*[, (docket 50277-240; OID-1997-048-06PRO)];

Page 1, lines 16-20:

U.S. Provisional Patent Application Serial No. 60/185,139 entitled METHOD AND ARTICLE FOR MANAGING REFERENCES BETWEEN OBJECTS IN MEMORIES OF DIFFERENT DURATIONS IN A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton.[(docket 50277-257; OID-1997-048-13PRO)];

Page 1, lines 21-24:

U.S. Provisional Patent Application Serial No. 60/185,138 entitled STATIC OBJECT SYSTEM AND METHODOLOGY FOR IMPLEMENTING A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-215; OID-1997-048-09)];

Page 1, lines 25-27:

U.S. Provisional Patent Application Serial No. 60/185,134 entitled AURORA NATIVE COMPILATION, filed on February 25, 2000 by Dmitry Nizhegorodov [(docket 50277-324; OID-1997-048-18PRO)];

Page 2, lines 1-4:

U.S. Provisional Patent Application Serial No. 60/185,137 entitled ACCESSING SHORTER-DURATION INSTANCES OF ACTIVATABLE OBJECTS BASED ON OBJECT REFERENCES STORED IN LONGER-DURATION MEMORY, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-327; OID-1999-084-01PRO)];

Page 2, lines 6-9:

U.S. Provisional Patent Application Serial No. 60/185,135 entitled HANDLING CALLOUTS MADE BY A MULTI-THREADED VIRTUAL MACHINE TO A SINGLE THREADED ENVIRONMENT, filed on February 25, 2000 by Scott Meyer [(docket 50277-332; OID-1997-084-02PRO)].

Page 2, lines 13-15:

U.S. Patent Application Serial No. 09/248,295 entitled MEMORY MANAGEMENT SYSTEM WITHIN A RUN-TIME ENVIRONMENT, filed on February 11, 1999 by Harlan Sexton *et al.* [(docket 50277-178; OID-1997-048-01)];

Page 2, lines 16-19:

U.S. Patent Application Serial No. 09/248,291 entitled MACHINE INDEPENDENT MEMORY MANAGEMENT SYSTEM WITHIN A RUN-TIME ENVIRONMENT, filed on February 11, 1999 by Harlan Sexton *et al.* [(docket 50277-172; OID-1997-048-02)];

Page 2, lines 20-23:

U.S. Patent Application Serial No. 09/248,294 entitled ADDRESS CALCULATION OF INVARIANT REFERENCES WITHIN A RUN-TIME ENVIRONMENT, filed on February 11, 1999 by Harlan Sexton *et al.* [(docket 50277-179; OID-1997-048-03)];

Page 2, lines 24-26:

U.S. Patent Application Serial No. 09/248,297 entitled PAGED MEMORY MANAGEMENT SYSTEM WITHIN A RUN-TIME ENVIRONMENT, filed on February 11, 1999 by Harlan Sexton *et al.* [(docket 50277-261; OID-1997-048-17)];

Page 3, lines 1-3:

U.S. Patent Application Serial No. 09/320,578 entitled METHOD AND ARTICLE FOR ACCESSING SLOTS OF PAGED OBJECTS, filed on May 27, 1999 by Harlan Sexton *et al.* [(docket 50277-293; OID-1998-034-01)];

Page 3, lines 4-7:

U.S. Patent Application Serial No. 09/408,847 entitled METHOD AND ARTICLE FOR MANAGING REFERENCES TO EXTERNAL OBJECTS IN A RUNTIME ENVIRONMENT, filed on September 30, 1999 by Harlan Sexton *et al.* [(docket 50277-230; OID-1997-048-12)];

Page 3, lines 8-11:

U.S. Patent Application Serial No. 09/512,619 entitled METHOD FOR MANAGING MEMORY USING EXPLICIT, LAZY INITIALIZATION IN A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-214; OID-1997-048-07)];

Page 3, lines 12-15:

U.S. Patent Application Serial No. 09/512,622 entitled METHOD FOR MANAGING MEMORY USING ACTIVATION-DRIVEN INITIALIZATION IN A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-213; OID-1997-048-08)];

Page 3, lines 16-19:

U.S. Patent Application Serial No. 09/512,621 entitled SYSTEM AND METHODOLOGY FOR SUPPORTING A PLATFORM INDEPENDENT OBJECT FORMAT FOR A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-258; OID-1997-048-14)];

Page 3, lines 20-23:

U.S. Patent Application Serial No. 09/512,618 entitled METHOD AND APPARATUS FOR MANAGING SHARED MEMORY IN A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-259; OID-1997-048-15)]; and

Page 3, lines 24-27:

U.S. Patent Application Serial No. 09/512,620 entitled USING A VIRTUAL MACHINE INSTANCE AS THE BASIC UNIT OF USER EXECUTION IN A SERVER ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-403; OID-1997-048-19).]

Page 6, lines: 23-27 and Page 7, lines: 1-4:

Some of these techniques are described in greater detail in the co-pending, commonly assigned applications, U.S. Patent Application Serial No. [] 09/512,622 entitled METHOD FOR MANAGING MEMORY USING ACTIVATION-DRIVEN INITIALIZATION IN A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-213; OID-1997-048-08)] and U.S. Patent Application Serial No. [] 09/512,619 entitled METHOD FOR MANAGING MEMORY USING EXPLICIT, LAZY INITIALIZATION IN A RUN-TIME ENVIRONMENT, filed on February 25, 2000 by Harlan Sexton *et al.* [(docket 50277-214; OID-1997-048-07)], the contents of both of which are incorporated by reference in their entirety.

IN THE CLAIMS:

1. (Once Amended) A method for analyzing a program, comprising the steps of:
logging a plurality of stack traces and respective tags in a log file at respective points during
execution of the program; and
[marking one or more interesting tags within the log file] recording within the log file one or
more of the tags are one or more marked tags.

3. (Once Amended) The method according to claim 2, wherein the step of producing the report includes:

identifying one or more of the stack traces that are associated with any of the one or more [interesting] tags marked; and
producing the report based on the identified one or more of the stack traces.

4. (Once Amended) The method according to claim 2, wherein producing the report includes:

identifying a last stack trace that is associated with one of the one or more [interesting] marked tags; and
producing the report based on the identified [one or more of the] last stack trace[s].

5. (Once Amended) The method according to claim 1, wherein:

the tags indicate respective addresses of allocated objects; and

the one or more [interesting] marked tags indicate one or more respective addresses of migrated objects.

6. (Once Amended) A method for producing a diagnostic report for a program, comprising the steps of:

accessing a log file comprising a list of stack traces and respective tags at [respective] associated points during execution of the program and comprising one or more [interesting] of the recorded as one or more marked tags; and
producing the diagnostic report based on the log file.

7. (Once Amended) The method according to claim 6, wherein the step of producing the report includes:

identifying one or more of the stack traces that are associated with any of the one or more [interesting] marked tags; and
producing the report based on the identified one or more of the stack traces.

8. (Once Amended) The method according to claim 6, wherein producing the report includes:

identifying a last stack trace that is associated with one of the one or more [interesting] marked tags; and
producing the report based on the identified [one or more of the] last stack trace[s].

9. (Once Amended) The method according to claim 6, wherein:

the tags indicate respective addresses of allocated objects; and

the one or more [interesting] marked tags indicate one or more respective addresses of migrated objects.

10. (Once Amended) A computer-readable medium bearing instructions for analyzing a program, said instructions being arranged to cause one or more processors upon execution thereby to perform the steps of:

logging a plurality of stack traces and respective tags in a log file at respective points during execution of the program; and

[marking one or more interesting tags within the log file] recording within the log file one or more of the tags are one or marked tags.

12. (Once Amended) The computer-readable medium according to claim 11, wherein the step of producing the report includes:

identifying one or more of the stack traces that are associated with any of the one or more [interesting] marked tags; and

producing the report based on the identified one or more of the stack traces.

13. (Once Amended) The computer-readable medium according to claim 11, wherein producing the report includes:

identifying a last stack trace that is associated with one of the one or more [interesting] marked tags; and
producing the report based on the identified [one or more of the] last stack trace[s].

14. (Once Amended) The computer-readable medium according to claim 10, wherein:

the tags indicate respective addresses of allocated objects; and

the one or more [interesting] marked tags indicate one or more respective addresses of migrated objects.

15. (Once Amended) A computer-readable medium bearing instructions for producing a diagnostic report for a program, said instructions being arranged to cause one or more processors upon execution thereby to perform the steps of:

accessing a log file comprising a list of stack traces and respective tags at [respective] associated points during execution of the program and comprising one or more [interesting] marked tags; and

producing the diagnostic report based on the log file.

16. (Once Amended) The computer-readable medium according to claim 15, wherein the step of producing the report includes:

identifying one or more of the stack traces that are associated with any of the one or more [interesting] marked tags; and
producing the report based on the identified one or more of the stack traces.

17. (Once Amended) The computer-readable medium according to claim 15, wherein producing the report includes:

identifying a last stack trace that is associated with one of the one or more [interesting] marked tags; and
producing the report based on the identified [one or more of the] last stack trace[s].

18. (Once Amended) The computer-readable medium according to claim 15, wherein:

the tags indicate respective addresses of allocated objects; and
the one or more [interesting] marked tags indicate one or more respective addresses of migrated objects.

19. (New) The method according to claim 4, wherein the step of producing the report includes:

processing the log file from the end backward until the beginning.

20. (New) The method according to claim 8, wherein the step of producing the report includes:

processing the log file from the end backward until the beginning.

21. (New) The computer-readable medium according to claim 13, wherein the step of producing the report includes:

processing the log file from the end backward until the beginning.

22. (New) The computer-readable medium according to claim 18, wherein the step of producing the report include:

processing the log file from the end backward until the beginning.